

IN THE CLAIMS

Please amend the claims as follows.

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Currently Amended) A computer-implemented compilation environment, comprising:
a computer system;
a software compiler run on the computer system to compile a source code into a compiled code;
a software linker run on the computer system to link the compiled code with other separately compiled codes into an executable program;
an optimizing analyzer within the compiler and also run on the computer system to generate a first optimizing transformation and a second optimizing transformation and their satisfying conditions for the compiled code without knowing information regarding the other separately compiled codes at compile-time; and
an optimization transformation module within the linker and also run on the computer system to determine, at link-time and with knowledge of the information of the other separately compiled codes, which of the first and second optimizing transformations should be selected

when the compiled code is linked with the other separately compiled codes, and to execute the selected one of the first and second optimizing transformations at link-time.

8. (Original) The compilation environment of claim 7, wherein the optimization transformation module determines which of the first and second optimizing transformations should be selected by checking the satisfying conditions with information only available at link-time.

9. (Original) The compilation environment of claim 7, wherein the first and second optimizing transformations and their satisfying conditions are in the form of conditional relocation operations.

10. (Original) The compilation environment of claim 7, wherein the first and second optimizing transformations are selected from a group comprising address base binding optimizing transformations, function cloning optimizing transformations, and data allocation optimizing transformations.

11. (Currently Amended) A method of performing code optimization, comprising
generating, at compile-time, a first optimizing transformation and a second optimizing transformation and their satisfying conditions for a compiled code which is to be linked with other separately compiled codes into an executable program at link-time, wherein generating the first and second optimization transformations and their satisfying conditions is done without knowing information regarding the other compiled codes at compile-time;
determining, at link-time and with knowledge of the information of the other separately compiled codes, which of the first and second optimizing transformations should be selected when the compiled code is linked with other compiled codes; and
executing the selected one of the first and second optimizing transformations at link-time.

12. (Original) The method of claim 11, wherein the first and second optimizing transformations and their satisfying conditions are generated in the form of conditional relocations.
13. (Original) The method of claim 11, wherein determining which of the first and second optimizing transformations should be selected further comprises checking the satisfying conditions with information only available at link-time to determine which of the first and second optimizing transformations should be selected.
14. (Original) The method of claim 11, wherein the first and second optimizing transformations are address base binding optimizing transformations.
15. (Original) The method of claim 11, wherein the first and second optimizing transformations are data allocation optimizing transformations.
16. (Original) The method of claim 11, wherein the first and second optimizing transformations are function cloning optimizing transformations.
17. (Currently Amended) An article of manufacture comprising a machine accessible medium including sequences of instructions, the sequences of instructions including instructions which, when executed, cause the machine to perform:
- generating, at compile-time, a first optimizing transformation and a second optimizing transformation and their satisfying conditions for a compiled code which is to be linked with other separately compiled codes into an executable program at link-time, wherein generating the first and second optimization transformations and their satisfying conditions is done without knowing information regarding the other compiled codes at compiler-time;
- determining, at link-time and with knowledge of the information of the other separately compiled codes, which of the first and second optimizing transformations should be selected when the compiled code is linked with other compiled codes; and
- executing the selected one of the first and second optimizing transformations at link-time.

18. (Original) The article of manufacture of claim 17, wherein the first and second optimizing transformations and their satisfying conditions are generated in the form of conditional relocations.
19. (Original) The article of manufacture of claim 17, wherein determining which of the first and second optimizing transformations should be selected further comprises checking the satisfying conditions with information only available at link-time to determine which of the first and second optimizing transformations should be selected.
20. (Original) The article of manufacture of claim 17, wherein the first and second optimizing transformations are address base binding optimizing transformations.
21. (Original) The article of manufacture of claim 17, wherein the first and second optimizing transformations are function cloning optimizing transformations.
22. (Original) The article of manufacture of claim 17, wherein the first and second optimizing transformations are data allocation optimizing transformations.